A Highly Integrated 8mW H.264/AVC Main Profile Real-time CIF Video Decoder on a 16MHz SoC Platform

Huan-Kai Peng, Chun-Hsin Lee, Jian-Wen Chen, Tzu-Jen Lo, Yung-Hung Chang, Sheng-Tsung Hsu, Yuan-Chun Lin, Ping Chao, Wei-Cheng Hung, Kai-Yuan Jan

Department of Computer Science
National Tsing Hua University, Taiwan

Abstract - We present a hardwired decoder prototype for H.264/AVC main profile video. Our design takes as its input compressed H.264/AVC bit-stream and produces as its output video frames ready for display. We wrap the decoder core with an AMBA-AHB bus interface and integrate it into a multimedia SoC platform. Several architectural innovations at both IP and system levels are proposed to achieve very high performance at very low operating frequency. Running at 16MHz, our FPGA demo system is able to real-time decode CIF (352x288) video at 30 frames per second. Moreover, we take system cost into consideration such that only a single external SDRAM is needed and memory traffic minimized.

I Introduction

Video coding is essential to many applications such as MPEG-1 for VCD, MPEG-2 for DVD and DTV, and MPEG-4 for Video Over IP Network. Ever increasing semiconductor capacity makes possible more advanced coding methods that achieve better coding efficiency at the expense of more computation. H.264/AVC is the latest video coding standard jointly developed by ITU and MPEG Committee [5]. Since its introduction two years ago, we have observed intensive research and development activity worldwide.

Many approaches have been proposed for various H.264/AVC-based applications. They range from pure software implementation in CPU or DSP, multi-processor, Application-Specific Instruction-set Processor (ASIP), Co-processor to pure hardwired accelerator. For low power portable or very high-end applications, a hardwired coding/decoding engine is preferred. Since software portion is indispensable to support additional features such as error resilience and rate control, the hardware engine should be equipped with standard bus interface so that it can be integrated into embedded systems with different types of CPUs. Notable hardware accelerators in the open literature include NCTU [2], NTU [3], Conexant [4] and CCU [1].

Few previous works have considered system-level overhead resulting from communicating with various components such as storage devices, SDRAM, display controller, and CPU. In this paper, we analyze the whole system performance and optimize two main bottlenecks: system bus bandwidth and decoder accelerator throughput.

II. Proposed Architecture

Fig. 1 depicts the block diagram of our proposed decoder. It uses an AMBA-AHB bus interface to communicate with system components such as CPU for system control, SD card for bit-stream to be decoded, SDRAM for reference frames and display buffering, and display controller. Our decoder core employs a four-stage pipelined architecture: (1) Bit-stream parser and Entropy decoding including both Context Adaptive Variable Length Decoding (CAVLD) and Context Adaptive Binary Arithmetic Decoding (CABAD), (2) Inverse Quantization/Discrete Cosine Transformation (IQ/IDCT) and Compensation including inter-frame Motion Compensation (MC) and first half of intra-frame Prediction (Intra Pred), (3) Picture Reconstruction and second half of Intra Pred, and (4) Deblocking Filter (DF). The Memory Fetch Unit (MFU) between the AMBA-AHB bus and the accelerator manages all external memory accesses from bit-stream parser, MC, and DF.

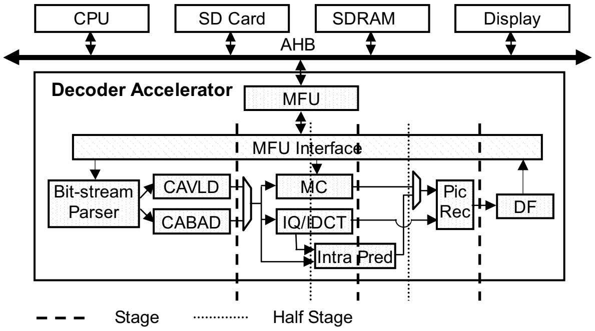

Fig. 1. Proposed H.264/AVC Decoding System

III. Performance Optimization

We propose two major optimizations: flexible pipeline scheduling and external memory traffic reduction. For the former, because each component in the pipeline takes different number of cycles for different types of video data, we can shut down some of them at appropriate time to save energy consumption. For example, MC and Intra Pred will never be activated in the same time neither are CAVLD and CABAD. Furthermore IQ/IDCT may be terminated early in case of encountering a SKIP mode.

External memory access has a major impact on the performance of the whole system and usually does not receive enough attention. The top-most part of Fig. 2 illustrates that our system needs to run at 112MHz to do real-time decoding if its external memory access is not scheduled. It is because a DRAM requires 5~20 cycles to initialize any sequential access, called a burst initial. To

1-4244-0630-7/07/$20.00 ©2007 IEEE.

make use of this property, our MFU buffers all external memory accesses at MB level, and issues them in a more sequential order from the DRAM controller's point of view. This technique greatly reduces the number of required burst initials, and has been shown effective as depicted in the middle part of Fig. 2. Observe that the "reference frame read" access now occupies over 55 percent of the whole external memory traffic. However, the reference frame read by the current MB can overlap with that of the previous one by 2/3 as shown in Fig. 3(b). Reusing these overlapped data, together with rearranging the reference frame format as shown in Fig. 3(a), leads to only four (16 pixels per MB/ 4 pixels per 32-bit data) burst initials needed per reference frame fetching of a MB. The performance boost of this scheme is shown in the bottom part of Fig. 2 at the expense of 42K logic gates and 9KB of internal memory.

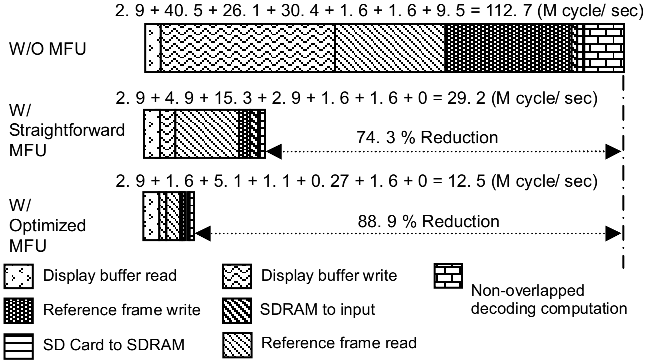

Fig. 2. Number of Required Bus Cycles for 30fps Decoding with Different Memory Fetching Schemes

IV. Experimental Results

We have implemented our proposed decoder using Verilog HDL and synthesized it using both Design Compiler targeted toward a TSMC 130nm CMOS cell library and Altera Quartus II targeted towards an FPGA libraries as shown in TABLE I. For the 120MHz ASIC version, our decoder takes 232K logic gates and 21KBytes of SRAM memory. At system-level, our system can demonstrate CIF 30fps decoding on an FPGA platform operated in 16MHz as shown in Fig. 4. Using PrimePower to analyze the ASIC version running at 16MHz, the power consumption is only 8 mW.

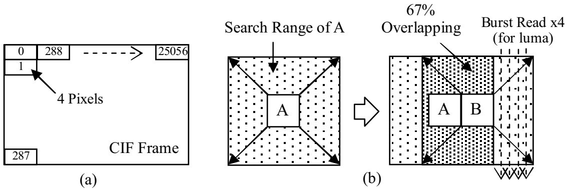

Fig. 3. (a) Optimal Format of Reference Frame (b) The Overlapping between Two Neighboring MBs

V. Summary and Conclusions

We have described an FPGA prototype of H.264/AVC main profile video decoder. Emphasizing on both accelerator performance and system memory traffic reduction, we are able to demonstrate a real-time decoder running at only 16MHz.

In the future, we would like to further improve the energy efficiency of our design as well as enhance its performance for super-HDTV applications (1920x4 X 1080x4).

TABLE I. Synthesis Results

Frequency	120MHz	16MHz	
Technology	TSMC 130nm	Altera FPGA	
IP	Gate Count	#LUTs	#Reg bits
Parser	20K	4,413	588
CABAD	34K	7,335	2,030
CAVLD	16K	3,814	637
Intra Pred	20K	7,104	575
MC	46K	5,449	2,230
IQ/IDCT	36K	5,781	1,188
Pic Rec	2K	375	274
DF	20K	3,637	1,407
Control Logic	2K	793	93
MFU	42K	5012	1100
Total	232K	43,538	8,239

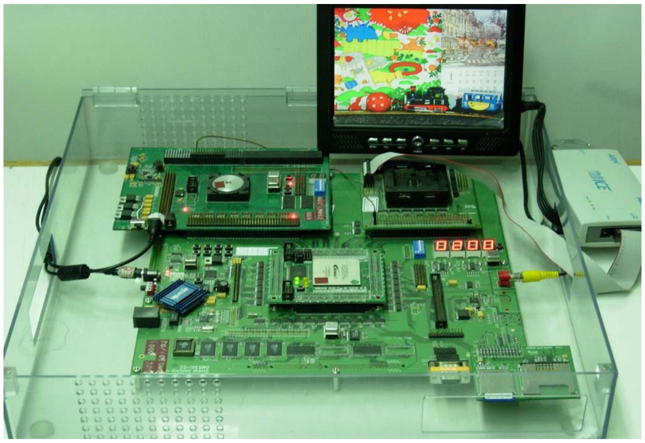

Fig. 4. Real-time H.264/AVC Decoding Demonstration

References

[1] C. C. Lin, et al.," A 160k Gate 4.5Kb SRAM H.264 Video Decoder for HDTV Applications," Proc. ISSCC, Feb., 2006

[2] T. A. Lin, et al.," An H.264/AVC Decoder with 4x4 Block-Level Pipeline," Proc. ISCAS, pp. 1810-1813, May, 2005

[3] T. W. Chen, et al.," Architecture Design of H.264/AVC Decoder with Hybrid Task Pipelining for High Definition Videos," Proc. ISCAS, pp. 2931-2934, May, 2005

[4] Y. Hu, A. Simpson, K. McAdoo, and J. Cush," A High Definition H.264/AVC Hardware Video Decoder Core for Multimedia SoCs," Proc. ISCE, pp. 385-389, Sept., 2004.

[5] Draft ITU-T Recommendation and Final Draft international Standards of Joint Video Specification (ITU-T Rec. H.264\ISO/IEC 14 496-10 AVC) Joint Video Team (JVT), Doc. JVT-G050, Mar., 2003.